

Functional Dependency Maintenance and Lossless Join

Decomposition in XML Model Decomposition

Li xia, Ye Fei-yue, Yuan Hong-juan, Peng Wen-tao

School of Computer Engineering and Science, Shanghai University, Shanghai
sdyylx112@sohu.com

Abstract

The paper studies “equality” in XML model decomposition. XML IFD, TFD, AFD and MVD are proposed. According to complexity of these FDs, four XML normal forms are presented. FD maintenance and lossless join decomposition of DTD are defined to analyze “equality” of decomposition. Four lossless algorithms are proposed to decompose DTD into some XML normal form and validity of these algorithms is analyzed.

Keywords: XML FD; DTD Normalization; XML Normal Forms

1. Introduction

XML is an emerging standard for exchange, representation and integration of internet data. In semantic grid^[1], SLN-Builder and Intelligent Semantic Browser are connected through XML descriptions. A well designed DTD for XML document is needed in these applications. Normalizing XML model avoids diversified problems in data operations. To control data redundancy to some degree and ensure data validity, it is necessary to study XML normal forms to decompose XML model, reducing redundancy and keeping original semantics. At present, because of complex structure, there isn't a mature and perfect theory for XML normalization and model decomposition.

2. Related Works

XML FDs^[2] is presented in ORA-SS model to decrease data redundancy. But FDs can't denote relative constraints. [3] [4] give algorithms to normalize XML DTD. But they don't discuss some abnormal FDs and XML forms. Marcelo Arenas^[5] defines XML forms-XNF to eliminate data redundancy and update abnormality, but XML forms are limited to encoded XML document.

Based on the researches, the paper will define five XML FDs and four XML normal forms. Four algorithms are proposed to decompose DTD into some XML normal form. Validity of these algorithms is analyzed to maintain FD maintenance and lossless join decomposition.

3. XML Functional Dependency

XML FD exists in document tree hierarchy, which is relative or absolute. Based on DTD and tree model, [6] define XML FD, denoting relative constraints. Based on it, XML IFD, TFD, AFD and MVD are presented.

Definition 3.1 Indirect Functional Dependency (IFD)

Given $D(E, A, M, N, r)$. XML IFD is the form: $R_1, R_2(P_1/\sigma_1, \dots, P_k/\sigma_k \rightarrow P_y/\sigma_y) \in (D, \Sigma)^+$, if there is another FD: $R_1, R_2'(R_2/Q_1/\sigma_1, \dots, R_2/Q_n/\sigma_n \rightarrow P_y/\sigma_y) \in (D, \Sigma)^+$ and $[P_1/\sigma_1, \dots, P_k/\sigma_k] \rightarrow [R_2/Q_1/\sigma_1, \dots, R_2/Q_n/\sigma_n] \notin (D, \Sigma)^+$, $[R_2/Q_1/\sigma_1, \dots, R_2/Q_n/\sigma_n]$

$\sigma_n \rightarrow [P1/\sigma_1, \dots, Pk/\sigma_n] \notin (D, \Sigma)^+, \forall P \in [P1/\sigma_1, \dots, Pk/\sigma_k]$ and $P \notin [Q1/\sigma_1, \dots, Qn/\sigma_n], \exists i \in [1, n], P_y \supseteq_{\text{path}} P_i, R2 \supseteq_{\text{path}} R2', \sigma$ is attributes of elements.

Definition 3.2 Transitive Functional Dependency (TFD)

Given $D(E, A, M, N, r)$. XML TFD is the form: $R1, R2(Q1/\sigma_1, \dots, Qn/\sigma_n \rightarrow P1/\sigma_1, \dots, Pk/\sigma_k) \in (D, \Sigma)^+$, if there are $\varphi 1: R1, R2(Q1/\sigma_1, \dots, Qn/\sigma_n \rightarrow T1/\sigma_1, \dots, Tm/\sigma_m) \in (D, \Sigma)^+$ and $\varphi 2: R1, R2(T1/\sigma_1, \dots, Tm/\sigma_m \rightarrow P1/\sigma_1, \dots, Pk/\sigma_k) \in (D, \Sigma)^+$. At the same time $(R1, R2(T1/\sigma_1, \dots, Tm/\sigma_m \rightarrow Q1/\sigma_1, \dots, Qn/\sigma_n)) \notin (D, \Sigma)^+$ and $[P1/\sigma_1, \dots, Pk/\sigma_k] \notin [T1/\sigma_1, \dots, Tm/\sigma_m]$.

Definition 3.4 Abnormal Functional Dependency One (AFD1)

Given $D(E, A, M, N, r)$. If there is $\varphi: R1, R2(Q1/\sigma_1, \dots, Qn/\sigma_n \rightarrow P_y/\sigma_y)$, but $\varphi \notin (D, \Sigma)^+$, so AFD1 exists.

Definition 3.4 Abnormal Functional Dependency Two (AFD2)

Given $D(E, A, M, N, r)$. If there is $R1, R2(Q1/\sigma_1, \dots, Qn/\sigma_n \rightarrow P_y/@\sigma_y)$, but $(R1, R2(Q1, \dots, Qn \rightarrow P_y)) \notin (D, \Sigma)^+$, so AFD2 exists.

Definition 3.5 Multi-Functional Dependency (MVD)

Given $D(E, A, M, N, r)$. XML MVD is the form: $R1, R2(Q1, \dots, Qn \rightarrow P1, \dots, Pk)$. $R1$ is scale path or \mathcal{E} , $R2$ is target path, as subnodes of $R1$, $Q_i (1 \leq i \leq n)$ is head path and $P_j (1 \leq j \leq k)$ is body path. If $R1 = \mathcal{E}$, MVD is called absolute MVD, else relative MVD.

4. XML DTD Normalization

Normalization DTD is basic requirement. Four XML normal forms are presented to eliminate data redundancy step by step.

Definition 4.1 DTD Normalization

Given $D(E, A, M, N, r)$ and FD set Σ , if values of root, attribute and text nodes are atomic, D is called normalization DTD.

Definition 4.2 XML First Normal Form

Given $D(E, A, M, N, r)$ and FD set Σ , if (D, Σ) is normalized and there aren't IFDs in $(D, \Sigma)^+$, D is the first normal form (XNF-1).

Definition 4.3 XML Second Normal Form

Given $D(E, A, M, N, r)$ and FD set Σ , if (D, Σ) is normalized and there aren't IFDs and TFDs in $(D, \Sigma)^+$, D is the second normal form (XNF-2).

Definition 4.4 XML Third Normal Form

Given $D(E, A, M, N, r)$ and FD set Σ , if (D, Σ) is normalized and there aren't AFDs in $(D, \Sigma)^+$, D is the third normal form (XNF-3).

Definition 4.5 XML Forth Normal Form

Given DTD $D(E, A, M, N, r)$ and MVD set Σ , to a MVD, if value of Q_i can exclusively determines value of paths($\text{last}(Q_i)$), the DTD D is the forth XML normal form (XNF-4).

5. XML Model Decomposition

XML model decomposition maintains lossless and has "FD maintenance".

5.1 FD Maintenance of DTD

"FD maintenance" means that original FD relationship isn't lost during attributes division.

Definition 5.1.1 Given $D(E, A, M, N, r)$, FD set Σ and elements or attributes set $U, U_i \subseteq U$. FD set $\{R1, R2(Q1, \dots, Qn \rightarrow P1, \dots, Pk) \mid R1, R2(Q1, \dots, Qn \rightarrow P1, \dots, Pk) \in \Sigma^+, R1/R2/Q1, \dots, R1/R2/Qn, R1/R2/P1, \dots, R1/R2/Pk \subseteq U_i\}$ is called projection of Σ on U_i . It is marked $\coprod U_i(\Sigma)$.

Definition 5.1.2 Given $D(E, A, M, N, r)$, FD set Σ and elements or attributes set U , D is decomposed into $\{D1, D2, \dots, Dn\}$, marked p . If $\Sigma^+ = (\coprod D1(\Sigma) \cup \coprod D2(\Sigma) \cup \dots \cup \coprod Dn(\Sigma))^+$, p of D maintains FD.

5.2 Lossless Join Decomposition of DTD

Given DTD D can be expressed by relation [7].

Definition 5.2.1 Lossless Join Decomposition of DTD

Given $D(E, A, M, N, r)$ and relation model $R_D(R1, \dots, Rn)$, D' is transform of D through normalization rules. Corresponding to D' , there is relation model $R_{D'}(R1, \dots, Rn)$. If $R_{D'}$ is

lossless decomposition of R_D , D' is called lossless decomposition of D .

5.3 Normalization Algorithms

Four algorithms are presented to decompose XML DTD into XML normal forms.

Algorithm 5.3.1 Decomposition DTD into XNF-1

Input: DTD $D(E,A,M,N,r)$ and FD set Σ

Output: D' conformed to XNF-1

(1) initialization: $D'=D, \Sigma'=\Sigma$;

(2) for \forall IFD $\in \Sigma$ do

according to element heightened rule,

$\varphi_{new}=R1,R2'(R2/Q1/\sigma_1,\dots,R2/Qn/\sigma_n \rightarrow R$

$2/Py/\sigma_y); \Sigma'=(\Sigma'-\{\varphi\}) \cup \{\varphi_{new}\};$

(3) combine same element type in D' ;

(4) return (D').

Algorithm 5.3.2 Decomposition DTD into XNF-2

Input: DTD $D(E,A,M,N,r)$ and FD set Σ

Output: D' conformed to XNF-2

(1) initialization: $D'=D, \Sigma'=\Sigma$;

(2) for \forall TFD $\in \Sigma$ do

according to element creation rule, $\varphi_{new}=$

$R1,R2'(V_{new}/\sigma_1,\dots,V_{new}/\sigma_m \rightarrow V_{new}/\sigma$

$1,\dots,V_{new}/\sigma_k); \Sigma'=(\Sigma'-\{\varphi\}) \cup \{\varphi_{new}\};$

(3) combine the same element type in D' ;

(4) return (D').

Algorithm 5.3.3 Decomposition DTD into XNF-3

Input: DTD $D(E,A,M,N,r)$ and FD set Σ

Output: D' conformed to XNF-3

(1) initialization: $D'=D, \Sigma'=\Sigma$;

(2) for \forall AFDs $\in \Sigma$ do

if there is AFD1, according to element creation rule, $D'=D\{Py/\sigma_y:=R2/V_{new}(\sigma_1,\dots,\sigma_n)\}$

$\Sigma'=\Sigma\{Py/\sigma_y:=R2/V_{new}(\sigma_1,\dots,\sigma_n)\};$

if there is AFD2, according to attribute removal rule, $D'=D\{Qi/@\sigma_y:=Py/@\sigma_y\};$

$\Sigma'=\Sigma\{Qi/@\sigma_y:=Py/@\sigma_y\}.$

(3) combine same element type in D' ;

(4) return (D').

Algorithm 5.3.4 Decomposition DTD into

XNF-4

Input: DTD $D(E,A,M,N,r)$ and FD set Σ

Output: D' conformed to XNF-4

(1) initialization: $D'=D, \Sigma'=\Sigma$;

(2) for \forall non_flat MVD $\in \Sigma$ do

if($\text{last}(Qi) \neq \text{parent}(\text{last}(Pj))$), according to element heightened rule, last (Pj) is heightened to make it as subelement of last(Qi)

(3) combine the same element type in D' ;

(4) return (D').

5.4 Algorithm Analysis

5.4.1 FD Maintenance of Algorithms

Algorithm 5.3.1 and 5.3.4 apply element heightened rule and attributes of element aren't divided, keeping original FDs. Algorithm 5.3.2 adopts element creation rule, relation model $R(\sigma_1,\dots,\sigma_n,\sigma_1,\dots,\sigma_m,\sigma_1,\dots,\sigma_k, X)$ is decomposed into $R11(\sigma_1,\dots,\sigma_n,\sigma_1,\dots,\sigma_m, X)$ and $R12(\sigma_1,\dots,\sigma_m,\sigma_1,\dots,\sigma_k)$. In D corresponding to R , there is FD set $\Sigma^+ = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_1,\dots,\sigma_m)), (R1, R2(\sigma_1,\dots,\sigma_m \rightarrow \sigma_1,\dots,\sigma_k)), (R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X))\}$. $\prod D1(\Sigma) = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_1,\dots,\sigma_m)), (R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X))\}$; $\prod D2(\Sigma) = \{(R1, R2(\sigma_1,\dots,\sigma_m \rightarrow \sigma_1,\dots,\sigma_k))\}$; $(\prod D1(\Sigma) \cup \prod D2(\Sigma))^+ = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_1,\dots,\sigma_m)), (R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X)), (R1, R2(\sigma_1,\dots,\sigma_m \rightarrow \sigma_1,\dots,\sigma_k))\}$; There is $(\prod D1(\Sigma) \cup \prod D2(\Sigma))^+ = \Sigma^+$. So $D1$ and $D2$ are decompositions of D , keeping FDs. In algorithm 5.3.3: every AFD1 can be transformed according to element creation rule. R is decomposed into $R11(\sigma_1,\dots,\sigma_n, X)$ and $R12(\sigma_1,\dots,\sigma_n, \sigma_y)$. There is FD set $\Sigma^+ = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_y)), (R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X))\}$, corresponding to R in D . $D1$ and $D2$ are decompositions of D , corresponding to $R11$ and $R12$. $\prod D1(\Sigma) = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X))\}$; $\prod D2(\Sigma) = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_y))\}$; $(\prod D1(\Sigma) \cup \prod D2(\Sigma))^+ = \{(R1, R2(\sigma_1,\dots,\sigma_n \rightarrow \sigma_y)), (R1, R2(\sigma_1,\dots,\sigma_n \rightarrow X))\}$; There is $(\prod D1(\Sigma) \cup \prod D2(\Sigma))^+ = \Sigma^+$. So $D1$ and $D2$ keep FDs. To AFD2, according to attribute removal

rule, σ_y is removed from P_y to Q_i , keeping FDs of P_y and Q_i .

5.4.2 Lossless Join Decomposition

Algorithm 5.3.1 and 5.3.4 apply element heightened rule, keeping original elements or attributes. Algorithm 5.3.2 adopts element creation rule. Relation $R(\sigma_1, \dots, \sigma_n, \sigma_1, \dots, \sigma_m, \sigma_1, \dots, \sigma_k, X)$ is decomposed into $R11(\sigma_1, \dots, \sigma_n, \sigma_1, \dots, \sigma_m, X)$ and $R12(\sigma_1, \dots, \sigma_m, \sigma_1, \dots, \sigma_k)$. In algorithm 5.3.3: to AFD1, relation R is decomposed into $R11(\sigma_1, \dots, \sigma_n, X)$ and $R12(\sigma_1, \dots, \sigma_n, \sigma_y)$. R , $R11$ and $R12$ are respectively corresponding to D , $D1$ and $D2$. According to definition 5.2.1, $D1$ and $D2$ are lossless decompositions of D . To AFD2, according to attribute removal rule, attribute σ_y is removed from P_y to Q_i . D' is lossless decomposition of D .

5.4.3 Terminality of Algorithms

Four algorithms use element heightened, element creation and attribute removal rule repeatedly to transform given D . As long as there is a conversion in \sum , IFD, TFD, AFDs and MVD will reduce in number. As FDs are limited in \sum , four algorithms are terminable.

5.4.4 Time Complexity of Algorithms

Time complexity of four algorithms is determined by “for” cycle and combination. In every execution of “for” cycle, a FD will be taken out. At the worst thing it is supposed that all FDs are IFDs or TFDs or AFDs or MVDs in \sum and the number of FDs is n ($n=|\sum|$). So entire “for” cycle executes n times. Combination is to check up the same nodes. It is supposed that the number of nodes is m and $m=|D'|$, two elements is taken out every time and there are C_m^2 steps to combine the same elements. Time complexity of combination is $O(m^2)$ at the worst things. So time complexity of entire algorithm is $O(m^2+n)$.

6. Conclusions

The paper presents five XML FDs and four

XML normal forms. Four algorithms are presented to decompose DTD into XML normal form and validity of these algorithms is analyzed. As decomposition results have relations with chosen FDs, choosing FDs in witch sequence is optimized. Further research on the problem will be continued in the future.

7. References

- [1]H.Zhuge and R.Jia,et.al. “Semantic Link Network Builder and Intelligent Browser”, *Concurrency and Computation: Practice and Experience*, 16 (14) (2004), pp.1453 -1476.
- [2]M. Lee, T. Ling and W. Low, “Designing functional dependencies for XML”, the 8th International Conference on Extending Database Technology (EDBT), 2002, pp.124-141.
- [3]Tan ZJ, Shi BL, “Normalization for DTD” *Journal, Computer Research and Development*, 2004, 41(4):pp.594-601.
- [4]Zhang ZhP, Wang Chao and Zhu YY, “Constraint-Based Normalization Algorithm for XML Documents”, *Journal, Computer Research and Development*, 2005, 42(5):pp.755-764.
- [5]Marcelo Arenas and Leonid Lambkin. “A normal form for XML documents”, *Acme Symposium Principles of Database Systems (PODS)*, Madison, Wisconsin, USA, 2002.
- [6]Tan ZJ, Pang YM and Shi BL, “Reasoning about functional dependency for XML”, *Journal, Software*, 2003, 14(9):pp.1564-1570.
- [7] Dongwon Lee, Wesley W. Cbu, “Constraints preserving transformation from XML document type definition to relational schema”, the 19th Int,l Conf, Conceptual Modeling, Salt Lake, USA, 2000.

Xia Li, born in 1980. Since 2004, she is the graduated student of Shanghai University. Her research is database and data grid.

Fei-yue Ye, born in 1959. He is professor of Shanghai University. His main research are database, data grid and model identification.